Network Traffic Classification based on Single Flow Time Series Analysis

Josef Koumar[1,2], Karel Hynek[1], and Tomáš Čejka[1]

[1]*CESNET, a.l.e.*, Prague, Czech republic
Email: {koumar, hynekkar, cejkat}@cesnet.cz
[2]*Czech Technical University in Prague*, Czech republic
Email: koumajos@fit.cvut.cz

***Abstract*—Network traffic monitoring using IP flows is used to handle the current challenge of analyzing encrypted network communication. Nevertheless, the packet aggregation into flow records naturally causes information loss; therefore, this paper proposes a novel flow extension for traffic features based on the time series analysis of the Single Flow Time series, i.e., a time series created by the number of bytes in each packet and its timestamp. We propose 69 universal features based on the statistical analysis of data points, time domain analysis, packet distribution within the flow timespan, time series behavior, and frequency domain analysis. We have demonstrated the usability and universality of the proposed feature vector for various network traffic classification tasks using 15 well-known publicly available datasets. Our evaluation shows that the novel feature vector achieves classification performance similar or better than related works on both binary and multiclass classification tasks. In more than half of the evaluated tasks, the classification performance increased by up to 5 %.**

***Index Terms*—time series, unevenly spaced time series, time series analysis, classification, lomb-scargle periodogram, spectral analysis, network traffic, machine learning**

I. Introduction

Network traffic monitoring provides information about activities in a computer network—an essential insight for maintaining the service and its security. As the technology evolves, a classical approach using *Deep Packet Inspection* (*DPI*) is no longer feasible due to the increased privacy protection using encryption. Additional security features, such as the RFC draft *Encrypted Server Name Indication (ESNI)* [1], which encrypts even domain names, forces the development of new ways of monitoring and analysis to detect network threats and malicious activities.

Contrary to DPI, flow-based [2] monitoring uses only aggregated information and statistics about the communication—IP flows. The IP flow term is defined, e.g., by *Internet Protocol Flow Information Export (IPFIX)* specification as aggregated information about the sequence of packets observed within a specific timeslot with the same properties—usually IP addresses, transport protocol (often TCP or UDP), and ports. The most commonly used simple statistics are the sum of packets and the sum of bytes of the observed communication. Such representation of the traffic is universal enough to get a high-level overview of large networks with high volumes of traffic and even encrypted traffic.

Since flows contain mainly information from packet headers and do not extract the payload, they are not affected by the payload encryption and are the ideal candidate for encrypted traffic monitoring. Many research works [3]–[10] thus use it together with machine learning for encrypted traffic classification to increase visibility and identify encrypted malicious communication.

Nevertheless, simple statistics such as the sum of transferred bytes and packets do not usually carry enough information for reliable traffic classification. The information about individual packet sizes, which has been found extremely useful in previous research works [11]–[13], is lost in the packet aggregation into flow records. Therefore, several approaches to extend flows were proposed to increase the classification performance. For example, flows are often extended by the *Sequence of packet lengths and times (SPLT)* [11] or *Sequence of packet Burst Length and Time (SBLT)* [12], and application-specific information fields [14], [15].

The SPLT and SBLT sequences significantly increase the amount of information we can leverage for classification. Still, they cannot carry information about all packets transmitted in the flows for practical reasons such as limited memory of flow exporter or constrain on flow record size. Therefore, SPLT often contains only the first n packets from the flow. For example, the Cisco joy exporter[1] exports detailed information (packet size, timestamp, direction) up to the first 200 packets in a flow, ipfixprobe flow exporter[2] exports this information for only the first 30 packets.

Even these packet-extended flows thus still miss a lot of information when dealing with longer communications. Therefore, our approach proposes an additional feature set to extend IP flows with *Time Series Analysis (TSA)* to mitigate the information loss due to aggregation or limited SPLT or SBLT sequence size. Instead of extending flows for information about individual packets, we extend flows for 69 novel features and test them for network traffic classification. In our approach, we consider the flow as time series of network packets, i.e.,

This research was funded by the Ministry of Interior of the Czech Republic, grant No. VJ02010024: Flow-Based Encrypted Traffic Analysis and also by the Grant Agency of the CTU in Prague, grant No. SGS23/207/OHK3/3T/18 funded by the MEYS of the Czech Republic.

[1]https://github.com/cisco/joy
[2]https://github.com/CESNET/ipfixprobe

978-3-903176-59-1 ©2023 IFIP

Single Flow Time Series (SFTS). Using the analysis of SFTS, we generate a set of significant features, which describes time dependencies between packets, packet sequences, distribution of packets, and behavior of packets. We evaluate the usability and universality of the feature set using 23 different network classification tasks with 15 well-known public datasets and machine learning algorithms. Our evaluation showed that the novel feature vector achieves excellent classification performance, similar to or better than related works in both binary and multiclass classification tasks. In more than half of the evaluated tasks, the classification performance increased by up to 5 %.

Furthermore, we also performed feature reduction to enable the deployment on networks with size-constrained network telemetry (e.g., due to available bandwidth allocated for monitoring). Despite the decrease in available information, the reduced feature vector of only ten features still achieves very good performance and reduces the average classification accuracy (compared to the full feature vector of 69 features) by only 0.03%.

The main contributions of our work can be summarized as follows:

- We proposed a novel approach that uses Time Series Analysis to generate 69 novel features.
- We computed the proposed feature vector for 15 well-known network datasets and made them publicly available at Zenodo platform [16].
- Using the novel features, we designed network classifiers capable of multiple potential network threat detection using machine learning algorithms. The novel classifiers achieved excellent accuracy, exceeding the previous best results from relevant works. Threats include Botnet, Cryptomining, DoH, (D)DoS, Malicious DNS, Intrusion in IDS, IoT Malware, Tor, and VPN.
- Using the novel features, we designed several multiclass classifiers, which performed better than previously published state-of-the-art algorithms. The multiclass classification concerns Botnet, IDS, IoT Malware, Tor, and VPN.

This paper is divided as follows: Section II summarizes the related work of flow-based network traffic classification. Section III provides information about time series analysis concepts and describes a novel approach to time series analysis in the IP flow exporter. Section IV provides a complete description of features exported in the novel extended IP flow. Section VI describes the complete classification pipeline with classification results. Section VII concludes this paper.

II. Related works

Flow-based network classification is an important area with multiple challenging tasks and various approaches. The main constraint of the detection method lies in the input data and information extracted by the monitoring system. For example, the flow monitoring systems based on NetFlowV5[3] can export only basic statistics about the ongoing communication, significantly constraining the subsequent network detectors that often need additional data sources to maintain reasonable accuracy [17]. Thus, many proposals extend the basic flow records for various information. We can divide the flow extension into two main approaches: 1) Extension for packet sequences and 2) Extension for precomputed features.

A. Extension for raw packet information

The extension of flows for packet sequences embeds the raw packet-level information about ongoing connections into the flows. Typically, flows are extended for a sequence of packets lengths and times (SPLT) that can be directly used for classification as in the case of Luxemburk et al. [11], or can be additionally processed for additional feature extraction as in the case of [18].

Nevertheless, the SPLT sequence cannot contain data about all packets in the flows due to practical reasons. The larger flows require more processing power and consume more memory and bandwidth. Thus the ipfixprobe flow exporter limits the size of the SPLT sequence to 30 packets.

To capture information about the packets that do not fit into the SPLT, researchers extend flows for additional features that we consider raw. For example, Tropkova et al. [12] proposed to use a Sequence of Burst lengths and Times (SBLT), which carries the information about individual packet bursts (times, amount of transferred data). Nevertheless, even SBLT has its length limit. Moreover, the aggregation of packets into the bursts loses some information about the exact timing of packets inside the burst.

B. Extension for precomputed features

Instead of exporting raw packet data that can be then processed by additional feature extraction, this approach computes the statistical features inside the exporter itself. An example of such an exporter is the CICFlowMeter[4] that extends each flow with 80 statistical features—mainly mean, standard deviation, max, and min of multiple countable information from packets, such as the number of packets and bytes. These features are then used by multiple researchers in various network classification tasks [19]–[23].

Similarly, as CICFlowMeter, MontazeriShatoori et al. [24] created a DoHLyzer exporter[5] that produces features directly within the flows. Nevertheless, the feature vector is entirely different from the features supported by the CICFlowMeter.

Compared to the SPLT and other raw-packet flow extensions, the computation of features directly in the exporter can capture statistical information across the whole flow, and no packet is missed. Nevertheless, it also aggregates packets. The packet aggregation then causes information loss, especially in the timing domain, which is not properly captured by the existing exporters and their feature extraction capability. However, time-related features such as periodicity are essential in

[3]https://www.cisco.com/c/en/us/td/docs/net_mgmt/netflow_collection_engine/3-6/user/guide/format.html

[4]https://github.com/ahlashkari/CICFlowMeter

[5]https://github.com/ahlashkari/DoHLyzer

the network classification, as shown by Koumar et al. [25] or MontazeriShatoori et al. [24].

In this work, we focus on the IP-flow extension for precomputed features to capture information about all packets. Compared to previous approaches, we aimed to create a universal feature vector that contains features based on statistical, time, distribution, frequency, and behavior properties acquired from the Time Series Analysis of the packet time series of each flow, i.e., SFTS [26]. Moreover, compared to all previous approaches, the universality and usefulness of the feature vector have been verified on 23 different network classification tasks using 15 network datasets.

III. Time Series Analysis in the IP Flow Exporter

This section describes several terms to explain our approach to time series analysis in the flow exporter to create a novel flow extension. We consider two crucial times for exporting flows—the inactive timeout is set to 65 seconds, and the active timeout is set to 300 seconds[6]. These settings belong to the open source IP flow exporter *ipfixprobe*.

In the state of the art of analysis of the time series from network traffic are mostly time series considered with evenly spaced time between observations. This type of time series is called evenly spaced or regularly sampled, and it is defined as the sequence of observation $\{X_n\} = \{x_1, \ldots, x_n\}$ taken in times $\{T_n = t_1, \ldots, t_n\}$, where n is the number of observations and is always true: $t_{j+1} - t_j = t_j - t_{j-1}, \forall j \in 2, \ldots, n$. Because of this behavior, it is possible to apply subtraction and division and get the sequence of times $\{T_n\} = 1, 2, \ldots, n$. So when an evenly spaced time series is used, then it is written only as $\{X_n\}$ where $n := 1, 2, \ldots, n$ and absolute observation times are unnecessary.

It is possible to use evenly spaced time series to analyze network traffic, mainly for forecasting and anomaly detection. Some previous works [8] use evenly spaced time series even for classification. However, network traffic naturally occurs with unevenly spaced timestamps (packet transmission time). Moreover, to create an evenly spaced time series, we need to set the aggregation interval—the time window for a single datapoint in the series—that highly affects the analysis result due to packets occurring at the aggregation interval borders. Badly selected aggregation intervals then cause analysis failure. Unfortunately, each time series has a different ideal aggregation interval—thus, the analysis failure with evenly spaced time series is (for some time series) inevitable [26].

In our approach, we create time series from packets within a flow—the series payload sizes in bytes with the corresponding transmission timestamp to create a time series. We call them Single Flow Time Series (SFTS). However, the SFTS created by the sizes of packets and their timestamps do not have evenly spaced timestamps between the datapoints. That means a time series of observations $\{X_n\} = (x_1, x_2, \ldots, x_n)$ taken at times $\{T_n\} = (t_1, t_2, \ldots, t_n)$ does not have constant $\delta_j = t_{j+1} - t_j, \forall j \in \{1, \ldots, n-1\}$. This type of time series is called unevenly (or unequally/ irregularly) spaced.

[6]If no packet is observed within the "inactive timeout" period, the flow is considered terminated. Flows longer than the "active timeout" are split and are exported every time this timeout elapses.

IV. Features description

This section contains a detailed description of novel time series features. We organized the features into five categories: 1) statistical, 2) time-based, 3) frequency-based, 4) distribution based, and 5) behavioral. Some of our proposed features for network classification were already used for classification in other fields of science, such as music classification [27]–[29]. The detailed description with mathematical equations of the whole feature set is published on the Zenodo platform [16].

A. Statistical-based features

The first set of features is based on statistical evaluation of the sequence of observation $\{X_n\}$ of the SFTS. The idea is a statistical description of data point deviation, i.e., statistical deviation of the packets' payload lengths. Table I shows the list of statistical-based features.

B. Time-based features

The time-based features describe the time axis of the unevenly-spaced time series $\{x_n\}$. For computation time-based features, we use a sequence of relative times $\{rt_n\} = t_i - t_0, i \in \{1, \ldots, n\}$, i.e., time from the beginning of a flow. Additionally, we use the sequence of time differences $\{dt_{n-1} = t_{i+1} - t_i, i \in \{1, \ldots, n-1\}\}$, i.e., time spaces between packets. The set of time-based features that are exported in the extended flow is listed below:

Mean, median, 1^{st}, and 3^{rd} quartile of relative times features are computed from the relative times rt_n to capture the statistical properties of packet times.

Mean, median, min, and max of time differences features are statistics of the time differences dt_n and represent information about spaces in the SFTS of the flow.

Duration is the last data point in the relative times rt_n.

C. Distribution-based features

The set of distribution-based features that are exported in the extended flow describes the distribution of data points in the SFTS $\{x_n\}$. The distribution features are listed below:

Hurst exponent can identify three behavior of time series—long-term switching between high and low values, long-term autocorrelation, and random (uncorrelated) time series.

Stationarity indicates the stationarity of the time series.

Benford's law computes the probability of satisfaction of Benford's law for occurrence counts of the nine most frequent packet lengths.

Normal distribution captures the probability that the SFTS is distributed by the normal distribution.

Count distribution captures the packet distribution within the SFTS—if the majority of data was sent at the beginning or at the end.

Count non-zero distribution is similar to feature *Count distribution* but filters the data points with zero value.

TABLE I
LIST OF STATISTICAL-BASED FEATURES

Feature	Feature	Feature	Feature	Feature	Feature	Feature
Mean	*Median*	*Standard deviation*	*Percent above mean*	*Fisher-Pearson G_1 skewness*	*Coefficient of variation*	*Kurtosis*
Variance	*Burtiness*	*First quartile*	*Percent below mean*	*Fisher-Pearson g_1 skewness*	*Pearson SK_1 skewness*	*Entropy*
Third quartile	*Min*	*Max*	*Min minus max*	*Fisher μ_3 skewness*	*Scaled entropy*	
Percent deviation	*Mode*	*Average dispersion*	*Root mean square*	*Pearson SK_2 skewness*	*Galton skewness*	

Time distribution describes the deviation of time differences between individual packets within the SFTS.

D. Frequency-based features

The idea of frequency-based features is to transform time series into the frequency domain and analyze it. Based on recent research [30]–[32], the frequency domain has several advantages over the time-domain. Frequency domain analysis is particularly useful for analyzing periodic behaviors because it allows analysis of the individual frequency components, and can be used to compare the frequency content of different time series. So, it is possible that we can get suitable features for the classification of network traffic from the frequency domain.

Since the SFTS are unevenly spaced, we must use the Lomb-Scargle (LS) periodogram [33] to transform the time series into a frequency domain. LS was originally developed for unevenly spaced time series in astrophysics.

The set of frequency-based features is listed below:

Min power, Max power features represent the minimum and maximum power of the LS periodogram.

Frequency of min power, Frequency of max power features describe the frequency of the minimum and maximum power of the LS periodogram.

Power mode, mean, stdev features describe the statistics of the power spectrum of the LS periodogram.

Spectral bandwidth describes the difference between upper and lower frequencies.

Spectral centroid indicates at which frequency the energy of a spectrum is centered upon.

Spectral energy represents the total energy present at all frequencies in LS periodogram.

Spectral entropy is the degree of randomness or disorder in the LS periodogram.

Spectral flatness estimates the uniformity of signal energy distribution in the frequency domain.

Spectral flux is the rate of change of periodogram power with increasing frequency.

Spectral kurtosis can indicate a non-stationary or non-Gaussian behavior in the power spectrum.

Spectral periodicity decides if in the LS periodogram is a significant peak that indicates the periodicity.

Spectral rolloff is defined as frequency bellow at is concentrated 85% of the distribution power.

Spectral spread is the difference between the highest and lowest frequency in the power spectrum.

Spectral skewness is the measure of peakedness or flatness of power spectrum.

Spectral slope is the slope of the power spectrum trend in a given frequency range.

Spectral zero crossing rate refers to the rate of power shifts, i.e., the change from negative to positive or the reverse.

E. Behavior-based features

The behavior-based features are focused on describing the specific set of behaviors of the SFTS. The set of behavior-based features that are exported in the extended flow is listed below:

Significant spaces indicates if there are some significantly bigger spaces between packets.

Switching ratio represents a value change ratio (switching) between payload lengths.

Transients indicates if a set of data points occurring in a short time window has significantly larger values.

Count of zeros represents a percentage of one-second intervals that do not contain any packets.

Biggest interval contains the maximal amount of data transferred in a one-second interval.

Directions describes a percentage ratio of packet direction.

Periodicity is the length and time of periodically occurring packet, if present.

V. DATASET SELECTION

We explored multiple publicly available datasets previously used or published in the network traffic classification domain. Nevertheless, a lot of datasets consist of already precomputed features and do not contain raw packet-based data, which is necessary for our feature extraction based on time-series analysis. Thus, we considered mainly the datasets where raw packet captures (PCAP files) were available. Together we selected 15 well-known network datasets that are written in Table II and processed them with our feature extraction. The processed datasets with our feature set were also published at Zenodo [16].

The selected datasets cover the most important traffic detection (binary) or classification (multiclass) tasks: 1) Botnet detection/classification, 2) Cryptomining detection, 3) DNS malware detection, 4) DNS over HTTPS detection, 5) DoS attack detection, 6) HTTPS Bruteforce detection, 7) Intrusion detection/classification, 8) IoT malware classification, 9) TOR detection/classification and 10) VPN traffic detection/classification.

In order to evaluate the performance of the novel features, we needed to create the baseline—a best-performing classifier for each concerned dataset. We searched for recent classifier

TABLE II
SUMMARIZED BEST-RELATED WORKS FOR CLASSIFICATION. IF THE "–" APPEARS, THEN THE RELATED WORKS DO NOT PRESENT THE METRICS, OR THE DATASET IS NOT DESIGNED FOR MULTICLASS CLASSIFICATION.

	Binary classification		
Detection problem	**Method**	**Accuracy**	**F1-score**
CTU-13 [34]	Stergiopoulos et al. [4]	99.85	99.90
CESNET-MINER22 [35]	Plný et al. [7]	93.72	90.59
CIC-Bell-DNS [20]	Kumaar et al. [36]	99.19	99.20
CIC-DoHBrw-2020 [8]	Zebin et al. [19]	99.98	99.91
DoH-Real-world [17]	Ježábek et al. [9]	97.5	98.7
HTTPS Brute-force [37]	Luxemburk et al. [10]	99.93	96.26
Bot-IoT [38]	Shafiq et al. [39]	99.99	99.99
Edge-IIoTset [40]	Khacha et al. [41]	99.99	99.99
IoT-23 [42]	Sahu et al. [43]	~ 96	~ 96
TON_IoT [44]	Dai et al. [45]	99.29	99.03
CIC-IDS-2017 [21]	Agrafiotis [22]	98.5	95.4
UNSW-NB15 [47]	Ding et al. [23]	92.39	94.39
ISCX-Tor-2016 [13]	Sarkar et al. [48]	99.89	99.88
ISCX-VPN-2016 [50]	Aceto et al. [51]	93.75	91.95
VNAT [53]	Jorgensen et al. [53]	–	98.00

	Multiclass classification		
Method	**Average**	**Accuracy**	**F1-score**
Marín et al. [5]	macro	99.72	76.04
	–		
	–		
	–		
	–		
	–		
	–		
Khacha et al. [41]	weighted	98.69	–
	–		
Tareq et al. [46]	weighted	98.5	98.57
Kunang et al.	weighted	95.79	95.11
Ding et al. [23]	macro	90.39	79.64
Yang et al. [49]	weighted	96.04	95.97
Dener et al. [52]	macro	89.29	87.83
Jorgensen et al. [53]	micro	96	96

proposals (published after 2017) using public research paper databases such as Google Scholar, IEEE Explore, and ACM Digital Library. We went through more than 300 papers and selected the best-performing proposals that met the following conditions ensuring fair comparability: 1) it was a flow-based method, 2) it uses the dataset as a whole and classifies all the dataset classes and types of samples, 3) does not use IP addresses as input features[7], 4) does not combine the concerned dataset with additional data. The selected best-performing proposals for both binary or multiclass versions of the classification tasks for each dataset are written in Table II.

VI. FEATURE EVALUATION

We evaluated the features by creating a novel classifier for each concerned network classification task. The classifier creation pipeline is the set of steps that creates the best final model. At first, the published datasets were split among Train, Validation, and Test sets in a ratio of 60:20:20 while keeping the labeling ratio like in the original datasets. Furthermore, some additional value sanitation is recommended, e.g., for a very short time series, it is required to handle "NaN" values: we replace NaN for the distribution features with 0.5, for the frequency features with -1, and for the rest of the features with 0. The source codes of our whole classification pipeline, including the pre-processing, are available at Github[8].

In the validation phase, we first select the optimal ML algorithm. We test 14 well-known ML algorithms such as Random Forest, K-NN, or SVM; nevertheless, the XGBoost algorithm achieved the best performance among all of the evaluated classification tasks. After the algorithm selection, we searched for the best model hyperparameters for optimal performance on each dataset without overfitting. We use the `hyperopt library` [56] to tune the following hyperparameters: `n_estimators`, `max_depth`, `gamma`, `reg_alpha`, `min_child_weight`, and `colsample_bytree`.

The hyperparameter search was performed using the training and validation datasets. The best values of the hyperparameters were selected based on the *F1-score* measure on the validation dataset. The final performance of the classifier on each dataset was obtained from the model trained using the trained part and evaluated on the test part. The test part was not used during any stage of the classifier design, ensuring the fairness of model evaluation on data that was not seen before.

A. Results

The results of binary and multiclass classification are presented in Table III. On most of the binary classification problems, the novel feature set achieved similar or better performance than the best-performing previous work. Moreover, our approach outperformed eight related works significantly (by more than 1%). However, on the TOR detection problem, we obtained a worse F1-score than the best classifier.

We investigated the differences between the TOR classifier published by Sarkar et al. [48] and found out that he uses a specially tailored feature vector that also includes transport ports. Even though transport ports are often used in network traffic classification, we intentionally opted to avoid them to maintain the universality of features. The classifier often tends to overfit the transport port features, which, in some cases, is not a desired behavior.

When we analyze the performance of the multiclass classification, we also outperformed most of the best-performing classifiers. Specifically, in five out of eight cases, we achieved more than a 1% classification performance increase. However, in two cases, we observed a slight decrease—TON_IoT and IDS-UNSW cases.

The best-performing classifier of TON_IoT published by Tareq et al. [46] is based on a 2D convolutional network (CNN) with very long packet-length data (SPLT with all

[7]The concerned datasets are mainly lab-created; thus usage of IP addresses is not considered—in this case—as a good practice due to dataset overfitting as described by Behnke et al. [54]

[8]https://github.com/koumajos/ClassificationBasedOnSFTS

TABLE III

FINAL RESULTS (IN %) OF CLASSIFICATION. THE GREEN-COLORED CELLS REPRESENT RESULTS WHERE OUR APPROACH IS SIGNIFICANTLY (BY 1% OR MORE) BETTER THAN BEST-RELATED WORK. CONTRARY THE RED-COLORED CELLS REPRESENT RESULTS WHERE OUR APPROACH IS SIGNIFICANTLY WORSE THAN BEST-RELATED WORK. FURTHERMORE, THE GRAY-COLORED CELLS REPRESENT RESULTS THAT ARE SIMILAR TO BEST-RELATED WORK.

	Binary classification			Multiclass classification			
Detection problem	**Method**	**Accuracy**	**F1-score**	**Method**	**Accuracy**	**Macro avg. F1**	**Weighted avg. F1**
Botnet	Stergiopoulos et al. [4]	99.85	99.90	Marín et al. [5]	99.72	76.04	
	Our approach	99.98	99.93	Our approach	**99.73**	**82.79**	**99.73**
Cryptomining	Plný et al. [7]	93.72	90.59		–		
	Our approach	**95.29**	**93.11**		–		
DNS Malware	Kumaar et al. [36]	99.19	99.20		–		
	Our approach	100.0	100.0		–		
DoH - CIC	Zebin et al. [19]	99.98	99.91		–		
	Our approach	99.90	99.84		–		
DoH - Real-world	Jeřábek et al. [9]	97.5	98.7		–		
	Our approach	97.79	98.80		–		
DoS	Shagiq et al. [39]	99.99	99.99		–		
	Our approach	100.0	100.0		–		
HTTPS Brute-force	Luxemburk et al. [10]	99.93	96.26		–		
	Our approach	**99.99**	**99.83**		–		
IDS - CIC	Agrafiotis [22]	98.5	95.4	Kunang et al. [55]	95.79	–	95.11
	Our approach	**99.89**	**99.75**	Our approach	**99.93**	**83.23**	**99.92**
IDS - UNSW	Ding et al. [23]	92.39	94.39	Ding et al. [23]	90.39	**79.64**	–
	Our approach	**98.49**	**98.50**	Our approach	**95.60**	40.22	95.08
IoT Mal. - Edge-IIoTset	Khacha et al. [41]	99.99	99.99	Khacha et al. [41]	98.69	–	–
	Our approach	99.99	99.97	Our approach	**99.97**	**89.75**	**99.97**
IoT Mal. - IoT-23	Sahu et al. [43]	~ 96	~ 96		–		
	Our approach	**99.86**	**99.91**		–		
IoT Mal. - TON_IoT	Dai et al. [45]	99.29	99.03	Tareq et al. [46]	98.5	–	**98.57**
	Our approach	**99.96**	**99.98**	Our approach	97.53	81.02	97.51
TOR	Sarkar et al. [48]	99.89	**99.88**	Yang et al. [49]	96.04		95.97
	Our approach	99.84	96.33	Our approach	95.48	79.87	95.20
VPN - ISCX	Aceto et al. [51]	93.75	91.95	Dener et al. [52]	89.29	87.83	
	Our approach	**94.35**	**95.48**	Our approach	**94.80**	**91.21**	**94.77**
VPN - VNAT	Jorgense et al. [53]	–	98.00	Jorgensen et al. [53]	96	–	–
	Our approach	**99.98**	**99.72**	Our approach	**98.60**	**98.88**	**98.60**

packets from connection) organized in the image. The SPLT data give the classifier advantage in the opportunity of high-quality feature extraction that allows accurate classification. Nevertheless, the long packet sequences (SPLT) cannot be exported in real-world deployment scenarios due to the technical limitations of the flow exporters (see Section II).

Besides, Ding et al. [23] achieved better results with IDS classification using IDS - UNSW dataset. According to our analysis, the better results are caused by a high-class imbalance of the dataset. Ding et al. thus proposed techniques for dealing with the imbalance ratio between classes. In our case, we did not deploy any imbalanced learn techniques in our classifier design pipeline to maintain comparability with the previous works—most of the concerned related works do not deploy any imbalanced learn techniques.

As can be seen in Table III, the proposed feature vector proved to be universal and performed well on all concerned tasks. The slight reduction in accuracy in some cases was expected since universal features cannot compete with specially tailored ones; however, we still consider the performance reduction, especially in the case of TOR or TON_IoT as a good tradeoff for universality and the possibility of deploying all the network classifiers behind single flow monitoring device.

VII. CONCLUSION

In this paper, we propose a novel feature set built from Time Series Analysis of Single Flow Time Series that can be used for classification methods. The proposed feature set is highly universal and achieves great results for both binary and multiclass classification. The feature set covers a wide range of behavior types in the following groups: 1) statistical deviation of payload lengths, 2) statistical deviation of packets times, 3) distribution of packets, 4) behavior of frequency domain, and 5) specific behaviors of data points. All groups contain significant features for classification.

The proposed method and feature set were evaluated on 23 network classification tasks using 15 publicly available and well-known network traffic datasets which are often used in recent research. All the collected datasets were processed to compute the proposed time series features that were published for any further research by the scientific community.

Overall, we trained and evaluated over 2,500 models across both binary and multiclass classification tasks and showed the universality of the proposed features. Furthermore, we prepared a prototype of the C++ implementation of the proposed feature vector extraction methods inside flow exporter ipfixprobe[9].

[9]https://github.com/koumajos/ipfixprobe_tsa_sfts

REFERENCES

[1] Eric Rescorla et al. TLS Encrypted Client Hello. Internet-Draft draft-ietf-tls-esni-16, Internet Engineering Task Force, 2023.

[2] Ganesh Sadasivan et al. Architecture for IP Flow Information Export. *RFC*, 5470:1–31, 2009.

[3] Miguel Leon et al. Comparative evaluation of machine learning algorithms for network intrusion detection and attack classification. In *IJCNN*, pages 01–08, 2022.

[4] George Stergiopoulos et al. Automatic Detection of Various Malicious Traffic Using Side Channel Features on TCP Packets. In *ESORICS 2018*, volume 11098, pages 346–362. Springer, 2018.

[5] Gonzalo Marín et al. Deep in the dark - deep learning-based malware traffic detection without expert knowledge. In *SPW 2019*, 2019.

[6] Wei Wang et al. Malware Traffic Classification Using Convolutional Neural Network For Representation Learning. In *ICOIN 2017*, pages 712–717. IEEE, 2017.

[7] Richard Plný et al. DeCrypto: Finding Cryptocurrency Miners on ISP Networks. In *NordSec 2022*. Springer, 2022.

[8] Mohammadreza MontazeriShatoori et al. Detection of doh tunnels using time-series classification of encrypted traffic. In *DASC/PiCom/CBDCom/CyberSciTech 2020*, pages 63–70. IEEE, 2020.

[9] Kamil Jerabek, Karel Hynek, Ondrej Rysavy, and Ivana Burgetova. Dns over https detection using standard flow telemetry. *IEEE Access*, 11:50000–50012, 2023.

[10] Jan Luxemburk et al. Detection of https brute-force attacks with packet-level feature set. In *CCWC 2021*, pages 0114–0122, 2021.

[11] Jan Luxemburk et al. Fine-grained TLS Services Classification With Reject Option. *Comput. Networks*, 220:109467, 2023.

[12] Zdena Tropková et al. Novel HTTPS Classifier Driven by Packet Bursts, Flows, and Machine Learning. In *CNSM 2021*. IEEE, 2021.

[13] Arash Habibi Lashkari et al. Characterization of Tor Traffic using Time based Features. In *ICISSP 2017*, pages 253–262. SciTePress, 2017.

[14] Martin Husák et al. Security Monitoring of HTTP Traffic Using Extended Flows. In *ARES 2015*. IEEE Computer Society.

[15] Min Hur et al. Towards smart phone traffic classification. In *APNOMS*, pages 1–4, 2012.

[16] Josef Koumar, Karel Hynek, and Tomáš Čejka. Network traffic datasets created by Single Flow Time Series Analysis, 2023.

[17] Kamil Jeřábek et al. Collection of datasets with DNS over HTTPS traffic. *Data in Brief*, 42:108310, 2022.

[18] Dmitrii Vekshin et al. DoH Insight: Detecting DNS Over HTTPS by Machine Learning. In *ARES 2020*. ACM, 2020.

[19] Tahmina Zebin et al. An explainable ai-based intrusion detection system for dns over https (doh) attacks. *IEEE Transactions on Information Forensics and Security*, 17:2339–2349, 2022.

[20] Samaneh Mahdavifar et al. Classifying Malicious Domains using DNS Traffic Analysis. In *DASC/PiCom/CBDCom/CyberSciTech 2021*, pages 60–67. IEEE, 2021.

[21] Iman Sharafaldin et al. Toward generating a new intrusion detection dataset and intrusion traffic characterization. *ICISSp*, 1:108–116, 2018.

[22] Georgios Agrafiotis et al. Image-based neural network models for malware traffic classification using pcap to picture conversion. In *ARES 2022*.

[23] Hongwei Ding et al. Imbalanced data classification: A knn and generative adversarial networks-based hybrid approach for intrusion detection. *Future Generation Computer Systems*, 131:240–254, 2022.

[24] Mohammadreza MontazeriShatoori et al. Detection of DoH Tunnels using Time-series Classification of Encrypted Traffic. In *DASC/PiCom/CBDCom/CyberSciTech 2020*, pages 63–70. IEEE, 2020.

[25] Josef Koumar et al. Network Traffic Classification Based on Periodic Behavior Detection. In *CNSM 2022*, pages 359–363. IEEE, 2022.

[26] Josef Koumar and Tomáš Čejka. Unevenly Spaced Time Series from Network Traffic. *TMA*, 2023. preprint on webpage at https://www.researchgate.net/publication/371530461_Unevenly_Spaced_Time_Series_from_Network_Traffic.

[27] Thomas L. Szabo. 5 - Transducers. In Thomas L. Szabo, editor, *Diagnostic Ultrasound Imaging*, Biomedical Engineering, pages 97–135. Academic Press, Burlington, 2004.

[28] Eric D. Scheirer et al. Construction and Evaluation of a Robust Multifeature Speech/Music Discriminator. In *ICASSP 1997*, pages 1331–1334. IEEE Computer Society, 1997.

[29] Alexander Lerch. *An Introduction to Audio Content Analysis: Applications in Signal Processing and Music Informatics*. Wiley-IEEE Press, 2012.

[30] A. Suarez et al. Analytical Comparison Between Time- And Frequency-domain Techniques for Phase-noise Analysis. *IEEE Transactions on Microwave Theory and Techniques*, 50(10):2353–2361, 2002.

[31] S. J Worley et al. Comparison of Time Domain and Frequency Domain Variables From the Signal-averaged Electrocardiogram: A Multivariable Analysis. *Journal of the American College of Cardiology*, 1988.

[32] Ralph Haberl et al. Comparison of Frequency and Time Domain Analysis of the Signal-averaged Electrocardiogram in Patients With Ventricular Tachycardia and Coronary Artery Disease: Methodologic Validation and Clinical Relevance. *Journal of the American College of Cardiology*, 12(1):150–158, 1988.

[33] Jacob T. VanderPlas. Understanding the lomb–scargle Periodogram. *The Astrophysical Journal Supplement Series*, 236(1):16, may 2018.

[34] S. García et al. An Empirical Comparison of Botnet Detection Methods. *Computers & Security*, 45:100–123, 2014.

[35] Richard Plný et al. *Datasets of Cryptomining Communication*. Zenodo, October 2022.

[36] M Kumaar et al. A Hybrid Framework for Intrusion Detection in Healthcare Systems Using Deep Learning. *Frontiers in Public Health*, 9, 2021.

[37] Jan Luxemburk et al. HTTPS Brute-force dataset with extended network flows, November 2020.

[38] Nickolaos Koroniotis et al. Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset. *Future Gener. Comput. Syst.*, 100:779–796, 2019.

[39] Muhammad Shafiq et al. Selection of Effective Machine Learning Algorithm and Bot-IoT Attacks Traffic Identification for Internet of Things in Smart City. *Future Gener. Comput. Syst.*, 107:433–442, 2020.

[40] Mohamed Amine Ferrag et al. Edge-iiotset: A new comprehensive realistic cyber security dataset of iot and iiot applications: Centralized and federated learning, 2022.

[41] Amina Khacha et al. Hybrid deep learning-based intrusion detection system for industrial internet of things. In *ISIA*. IEEE, 2022.

[42] Sebastian Garcia et al. IoT-23: A labeled dataset with malicious and benign IoT network traffic, January 2020.

[43] Amiya Kumar Sahu et al. Internet of things attack detection using hybrid deep learning model. *Computer Communications*, 176:146–154, 2021.

[44] Nour Moustafa. A new distributed architecture for evaluating ai-based security systems at the edge: Network ton_iot datasets. *Sustainable Cities and Society*, 72:102994, 2021.

[45] Jianbang Dai et al. Glads: A global-local attention data selection model for multimodal multitask encrypted traffic classification of iot. *Computer Networks*, 225:109652, 2023.

[46] Imad Tareq et al. Analysis of ton-iot, unw-nb15, and edge-iiot datasets using dl in cybersecurity for iot. *Applied Sciences*, 12(19):9572, 2022.

[47] Nour Moustafa et al. Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set). In *MilCIS*, pages 1–6. IEEE, 2015.

[48] Debmalya Sarkar et al. Detection of Tor Traffic using Deep Learning. In *AICCSA 2020*, pages 1–8. IEEE, 2020.

[49] Yang Yang et al. A network traffic classification method based on dual-mode feature extraction and hybrid neural networks. *IEEE Transactions on Network and Service Management*, 2023.

[50] Gerard Draper-Gil et al. Characterization of Encrypted and VPN Traffic Using Time-related. In *ICISSP*, pages 407–414, 2016.

[51] Giuseppe Aceto et al. DISTILLER: Encrypted Traffic Classification via Multimodal Multitask Deep Learning. *J. Netw. Comput. Appl.*, 2021.

[52] Murat Dener et al. Rfse-gru: Data balanced classification model for mobile encrypted traffic in big data environment. *IEEE Access*, 11:21831–21847, 2023.

[53] Steven Jorgensen et al. Extensible Machine Learning for Encrypted Network Traffic Application Labeling via Uncertainty Quantification. *CoRR*, abs/2205.05628, 2022.

[54] Matthew Behnke et al. Feature engineering and machine learning model comparison for malicious activity detection in the dns-over-https protocol. *IEEE Access*, 9:129902–129916, 2021.

[55] Yesi Novaria Kunang et al. Attack classification of an intrusion detection system using deep learning and hyperparameter optimization. *Journal of Information Security and Applications*, 58:102804, 2021.

[56] James Bergstra et al. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In *International conference on machine learning*, pages 115–123. PMLR, 2013.